数理解析研究所講究録 1375 巻 2004 年 171-173

直角二等辺三角形の二等分割とその頂点数について

三河 賢治 (Kenji MIKAWA)
新潟大学総合情報処理センター
mikawa@cc.niigata-u.ac.jp

長谷川 誠 (Makoto HASEGAWA)
新潟大学総合情報処理センター
hasegawa@cc.niigata-u.ac.jp

あらまし

近年，ディジタル画像データを幾つかの小領域に分割し，画像データを圧縮するアルゴリズムが活発に議論されている．例えば，画像データを逐次的に三角形に分割する圧縮の構造は二分木に対応する．画像を圧縮するアルゴリズムは，圧縮された画像データと原画像データのデータ量や画質を比較して，評価される．本研究では，ディジタル画像を幾つかの小三角形に分割し，画像データを圧縮するようなアルゴリズムを扱う．このようなアルゴリズムは，画像を三角形に分割する回数が画質に影響を与えるが，三角形分割された画像内の頂点数と画質との関係は明らかではない．そこで本研究では，三角形の頂点数と画質との関係を明らかにしていくための助走的な研究として，圧縮された画像内の頂点数を効率よく求めるアルゴリズムを提案する．

キーワード： 直角二等辺三角形の二等分割，二分木，組合せアルゴリズム，画像処理

1 まえがき

近年，ディジタル画像データを幾つかの小領域に分割し，画像データを圧縮するアルゴリズム [2, 4, 6] が活発に議論されている．本研究で扱う圧縮アルゴリズムは，始めに画像データを幾つかの直角二等辺三角形に分割し，次に各直角二等辺三角形を適当な閾値に基づいて逐次的に二等分割していく．各三角形の頂点には，原画像データの輝度値が与えられていて，頂点の輝度値から領域内部の画像データを近似する．このような圧縮のデータ構造は二分木で実現される．二分木の内部節点数と葉数は，それぞれ画像データを分割した回数と分割された小三角形の数に対応する．例として，ディジタル画像の三角形分割の構造を図 1 に示す．幾つかの小領域に分割するようなディジタル画像の圧縮は画像の拡大縮小に強く，自然画像のベクトル化 [1] に代表されるように様々な応用が提案されている．

画像を圧縮するアルゴリズムは，圧縮された画像データと原画像データのデータ量や画質を比較して，総合的に評価される．現在のところ，三角形の頂点数と画質との関係は明らかではなく，今後の解明に期待されている．そこで本研究は，三角形の頂点数と画質との関係を明らかにしていくための助走的な研究として，直角二等辺三角形を再帰的に二等分割した平面図形に対応する二分木を入力として与え，この平面図形の頂点数を効率よく求めるアルゴリズムを考えたい．任意の平面図形の頂点の数を v，辺の数を m，面の数を f とすると，Euler の多面体定理から

(a) Original image. (b) Triangulation.

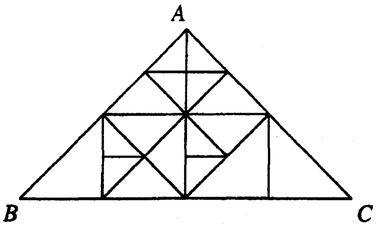
図 1 ディジタル画像と三角形分割の構造

$$v - m + f = 1$$

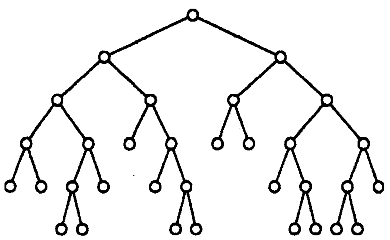
を満たすことが知られている．しかしながら，本研究で扱うような直角二等辺三角形の分割では，三角形を分割する箇所によって頂点数と辺数が変化するので Euler の関係式を直接利用することはできない．

2 準備

三角形の領域の分割と二分木との関係を次のように定義する．始めに，三角形の領域を二分木の節点 v に対応させる．次に三角形を分割する垂線について，斜辺を底辺とする垂線の左領域を v の左の子，右領域を v の右の子と定義する．根のみの二分木は，分割されていない三角形を



(a) Bisection triangulation.

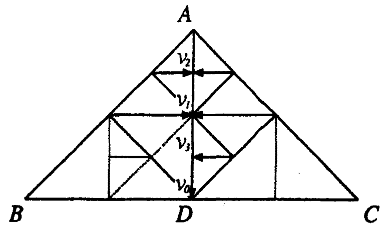


(b) Binary tree representation.

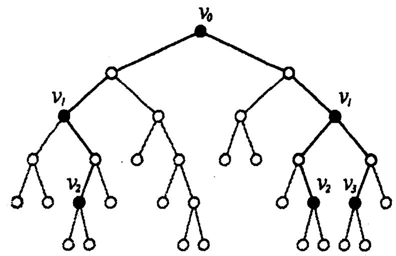
11110011000101010011001101001100

(c) Bitstring representation.

図 2 三角形の二等分割とその二分木表現



(a) Common vertices on bisection AD.



(b) Binary tree representation.

図 3 二等分線上の共有頂点と二分木表現

表す．三角形を分割する回数と分割された領域数は，それぞれ二分木の内部節点と葉に対応する．

二分木の構造を表す方法は幾つか知られているが，二分木の内部節点と葉にそれぞれ 1 と 0 を与え，行きがけ順に走査して得られるビット列で二分木を表現することにする．但し，最右の葉に与えられる 0 を省く．この方法によると，n 個の内部節点からなる二分木は長さ $2n$ のビット列となり，かつビット列の先頭から順に 0 と 1 の総数を比較すると常に 0 の総数は 1 の総数を超えない [5, 3]．例として，三角形の二等分割とその二分木表現，ビット列表現を図 2 に示す．

直角二等辺三角形であることに注目すると，三角形の領域の分割と頂点数との関係は次の二通りの場合に分けて考えるとよい．一つ目は三角形の領域を分割する毎に新しい頂点が一つ増える場合であり，二つ目は領域を分割しても頂点を共有するために新しい頂点が増えない場合である．前者の三角形の分割回数（即ち二分木の内部節点の数）を n，後者の共有する頂点数を c とすると，三角形の頂点数 N は次式

$$N = n - c + 3$$

で求めることができる．本研究では，三角形の分割の構造を示すビット列が入力値となるので，共有する頂点数 c を求めるアルゴリズムが重要になる．三角形の分割の定義から，ある三角形の領域が分割されると，次は分割された左右の小領域が再帰的に分割されていく．三角形を分割する二等分線に注目すると，次の性質を得る．

［性質 2.1］三角形の領域を分割する二等分線を線分と見なす．このとき三角形の領域を分割する二等分線は，他の再帰的に構成される三角形の領域を分割する二等分線とはならない．

3 探索アルゴリズム

共有頂点を求めるアルゴリズムの基本的な考え方は次の通りである．性質 2.1 を利用して，二等分線上で共有する頂点数を求めるアルゴリズムを構築し，分割された左右の小領域に対してこのアルゴリズムを再帰的に適用していく．そうすると，すべての共有する頂点を求めることができるだろう．例として，二等分線 AD 上で共有する頂点 v_1，v_2 と二分木表現との関係を図 3 に示す．頂点 v_3 は二等分線 AD 上の頂点であるが共有頂点ではない．

始めに，三角形の二等分線上の共有頂点を求めるアルゴリズムについて考えていく．左右の領域から二等分線上で初めて出会う頂点を v_1 とすると，次に二等分線上で出会う頂点は v_2 と v_3 の二つである．これらの頂点の関係を二分木上で考えると，左部分木の v_1 に対応する頂点から右の子を経由して左の子が v_2 に対応する頂点であり，v_1

```
procedure search(v_l, v_r)
begin
  if v_l ≠ leaf and v_r ≠ leaf then c := c + 1;
  L := leftchild(rightchild(v_l));
  R := rightchild(leftchild(v_r));
  if L ≠ leaf and R ≠ leaf then search(L, R);
  L := rightchild(leftchild(v_l));
  R := leftchild(rightchild(v_r));
  if L ≠ leaf and R ≠ leaf then search(L, R);
end;
```

図 4 Common vertices search.

```
procedure traversal(v)
begin
  if v ≠ leaf then
    L := leftchild(leftchild(v));
    R := rightchild(rightchild(v));
    if L ≠ leaf and R ≠ leaf then search(L, R);
    traversal(leftchild(v));
    traversal(rightchild(v));
  end;
end;
```

図 5 Binary tree traversal.

に対応する頂点から左の子を経由して右の子が v_3 に対応する頂点である．一方，右部分木の v_1 に対応する頂点から左の子を経由して右の子が v_2 に対応する頂点であり，v_1 に対応する頂点から右の子を経由して左の子が v_3 に対応する頂点である．線分 AD 上で共有頂点となる頂点は，ちょうど左部分木と右部分木で対象な経路をたどることによって得られる．このとき，左右部分木上で v_2（または v_3）に対応する頂点の少なくとも一方が葉である場合，二等分線 AD 上で共有しないので，v_2 から探索を開始しない．図 3 では，右部分木上の v_3 に対応する頂点は内部節点であるが，左部分木上の v_3 に対応する頂点は葉であるので，v_3 より後の共有頂点は存在しない．一方，左右部分木上で v_2 に対応する頂点はどちらも内部節点なので，v_2 を共有頂点として数え上げ，v_2 より下の経路に対して，再び上記の手続きを繰り返す．以上の探索方法を，二分木の任意の節点 v（すなわち任意の三角形領域）から始める手続き search を図 4 に示す．ここで，手続き search の二つの引数 v_l と v_r は，三角形領域 v の左領域から初めて共有頂点となる頂点に対応する二分木上の節点 v_l，右領域から初めて共有頂点となる頂点に対応する二分木上の節点 v_r である．一方，画像データに対応する二分木の根から順に訪問する手続きを図 5 に示す．

4 今後の展望

本研究では，二分木で表された直角二等辺三角形の共有頂点を求める素朴なアルゴリズムを提案した．今後の展開として，ビット列から直接共有頂点を求めるアルゴリズムや，計算量の小さいアルゴリズムの開発に取り組む予定である．

参考文献

[1] 長谷川誠，三河賢治，"デジタル濃淡画像の三角平面パッチによるSVG形式への変換，" 映情学誌，Vol.57，No.10，pp.1337–1341，Oct. 2003.

[2] 長谷川誠，山崎一生，"可変ブロック分割と双1次曲面パッチ近似による画像データの圧縮，" 信学論（D-II），Vol.J84-D-II，No.7，pp.1399–1408，Jul. 2001.

[3] I. Semba, "Generation of all the balanced parenthesis strings in lexicographical order," Inf. Process. Lett., Vol.12, No.4, pp.188–192, 1981.

[4] 李相善，田中弘美，"アダプティブメッシュを用いた画像の並列領域分割，" 信学論（D-II），Vol.J82-D-II，No.7，pp.1171–1179，Jul. 1999.

[5] F. Ruskey, T.C. Hu, "Generating Binary Trees Lexicographically," SIAM J. Comput., Vol.6, No.4, pp.745–758, 1977.

[6] 若月大輔，石井郁夫，髙橋章，今井博英，牧野秀夫，"VRオブジェクトの局部的な形状詳細度制御のためのマルチスケールパッチ生成法，" 信学論（D-II），Vol.J86-D-II，No.5，pp.697–705，May. 2003.